

Compte rendu TP Message secret sous ubuntu...



TP Linux : La Chasse au Fichier Secret
Missions de camouflage et de recherche pour administrateurs système en herbe.
Défi en binôme : créez et cachez un fichier secret, puis utilisez des stratégies de recherche pour retrouver celui de votre partenaire en temps limité.

MISSION 1 : CACHER LE SECRET (30 MIN)

L'objectif : Dissimuler un message
Le fichier doit être compressé, placé dans un dossier caché et protégé par des permissions.

- 1. Isoler le Secret**
Créer un utilisateur "gardien" dédié et une arborescence de dossiers cachés pour lui.
Icones: guardian, .hidden_folder, sub-user
- 2. Archiver et Protéger**
Compresser le fichier (tar.gz) et appliquer des permissions strictes (chmod 600/700).
Icones: file, tar.gz, lock, chmod 600/700
- 3. Brouiller les Pistes**
Supprimer les fichiers temporaires et ne laisser qu'un seul indice pour le chercheur.
Icones: broom, clue

MISSION 2 : TROUVER LE SECRET (30 MIN)

L'objectif : Mener l'enquête
Développer une méthodologie de recherche pour localiser le fichier en moins de 30 minutes.

- 1. Pister l'Utilisateur**
Analyser l'indice reçu et rechercher les utilisateurs "gardiens" suspects dans /etc/passwd.
Icones: magnifying glass, user, clue
- 2. Scanner le Système**
Utiliser la commande find pour localiser les dossiers cachés et les archives .tar.gz.
Icones: find, folder, archive
- 3. Examiner les Permissions**
Rechercher les fichiers et dossiers avec des droits d'accès inhabituels ou restreints.
Icones: magnifying glass, folder, permissions

Sommaire

Introduction.....	2
Mission 1 Cacher le mot secret.....	2
Mission 2 Trouver le mot secret.....	4
Conclusion.....	4

Introduction

Pour ce TP sous Ubuntu, le but était d'apprendre à sécuriser un système de manière ludique en défiant un binôme. J'ai dû imaginer et mettre en place une stratégie complète pour dissimuler un "message secret" (création d'utilisateur, gestion des permissions, suppression de traces), le défi étant de rendre ce fichier impossible à trouver ou à lire pour mon camarade.

Mission 1 Cacher le mot secret

1. J'ai créé le "Gardien" et son dossier caché

J'ai commencé par créer un utilisateur dédié nommé "gardien" et une arborescence contenant un dossier caché.

```
# J'ai créé le nouvel utilisateur 'gardien'  
sudo adduser gardien  
  
# Je me suis connecté en tant que ce nouvel utilisateur  
su - gardien  
  
# J'ai créé un dossier caché (le point devant le nom le rend invisible)  
mkdir .base_secrete
```

2. J'ai créé et archivé le secret

J'ai rédigé mon message, puis je l'ai compressé au format tar.gz pour l'isoler.

```
# J'ai créé le fichier contenant le message secret  
echo "Le code secret est 1234" > .base_secrete/message.txt  
  
# J'ai compressé le fichier (c=create, z=gzip, v=verbose, f=file)  
tar -czvf .base_secrete/archive_secrete.tar.gz .base_secrete/message.txt
```

3. J'ai protégé le fichier avec des permissions strictes

J'ai appliqué des droits d'accès restreints (chmod 600 ou 700) pour m'assurer que personne d'autre ne puisse lire l'archive.

```
# J'ai limité l'accès au fichier (lecture/écriture pour moi uniquement)  
chmod 600 .base_secrete/archive_secrete.tar.gz  
  
# J'ai sécurisé l'accès au dossier lui-même  
chmod 700 .base_secrete
```

4. J'ai brouillé les pistes

Pour terminer, j'ai supprimé le fichier texte original (fichier temporaire) et je n'ai laissé qu'un seul indice pour orienter la recherche.

```
# J'ai supprimé le fichier message original non compressé
rm .base_secrete/message.txt

# J'ai laissé un indice pour mon binôme
echo "Cherche l'archive cachée du gardien..." > indice.txt
```

5. J'ai supprimé les traces (l'historique des commandes)

Pour finir, j'ai effacé la liste des commandes que je venais de taper. Cela empêche mon binôme de retrouver le fichier secret simplement en regardant l'historique ou en appuyant sur la "flèche du haut". C'est l'étape ultime pour "brouiller les pistes".

```
# J'ai effacé l'historique de la session en cours
history -c

# J'ai vidé le fichier qui enregistre toutes les commandes
cat /dev/null > ~/.bash_history

# Je me suis déconnecté pour valider le nettoyage
exit
```

Mission 2 Trouver le mot secret

La mission 2 a été réalisé par mon binôme (il n'a pas réussi à trouver le mot secret car il lui a manqué du temps)

Conclusion

En conclusion, cet exercice m'a fait réaliser qu'il ne suffit pas de cacher un fichier pour le protéger, mais qu'il faut surtout bien verrouiller les droits d'accès. J'ai réussi ma mission de défense puisque mon binôme n'a pas trouvé mon fichier, ce qui m'a prouvé l'efficacité concrète des commandes de sécurité et du nettoyage d'historique que j'ai utilisées.